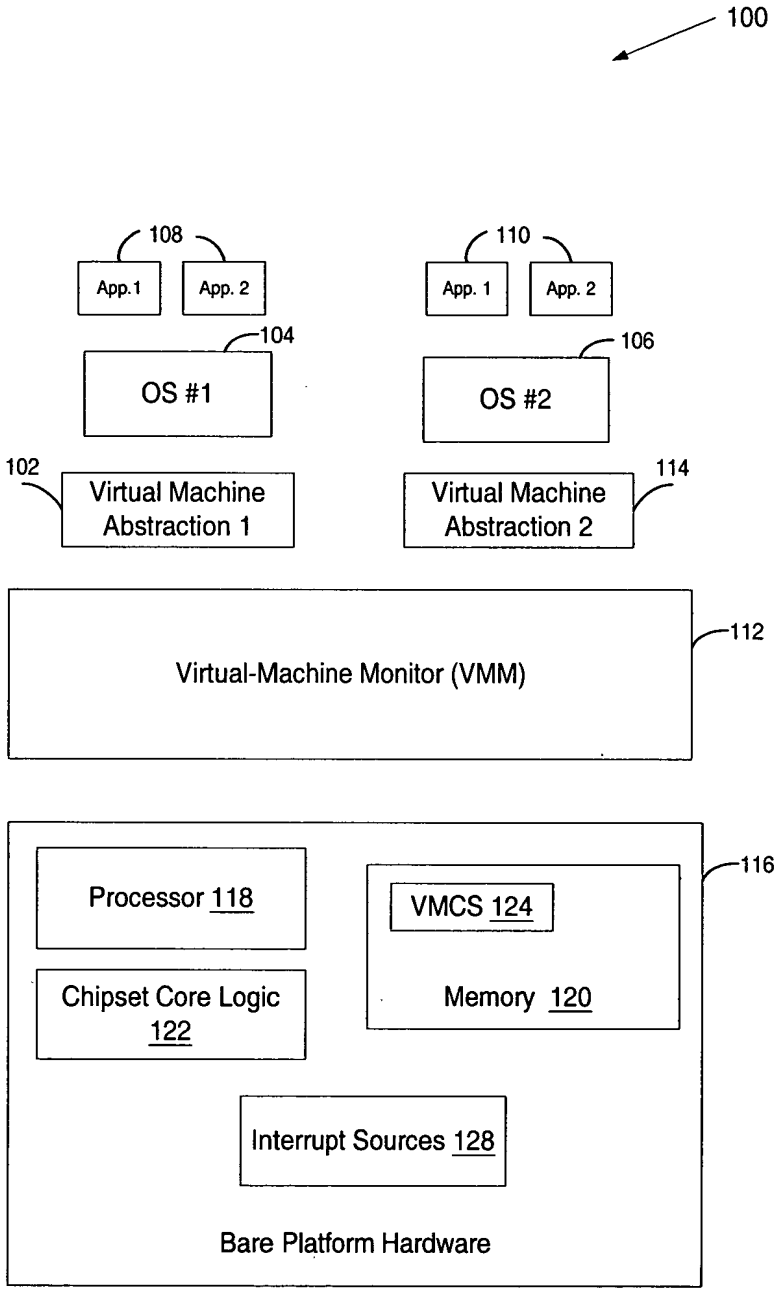
100
108
App.1
App. 2
110
App. 1
App. 2
104
OS #1
106
OS #2
102
Virtual Machine Abstraction 1
114
Virtual Machine Abstraction 2
Virtual-Machine Monitor (VMM)
112
Processor 118
VMCS 124
Chipset Core Logic 122
Memory 120
116
Interrupt Sources 128
Bare Platform Hardware

FIG. 1

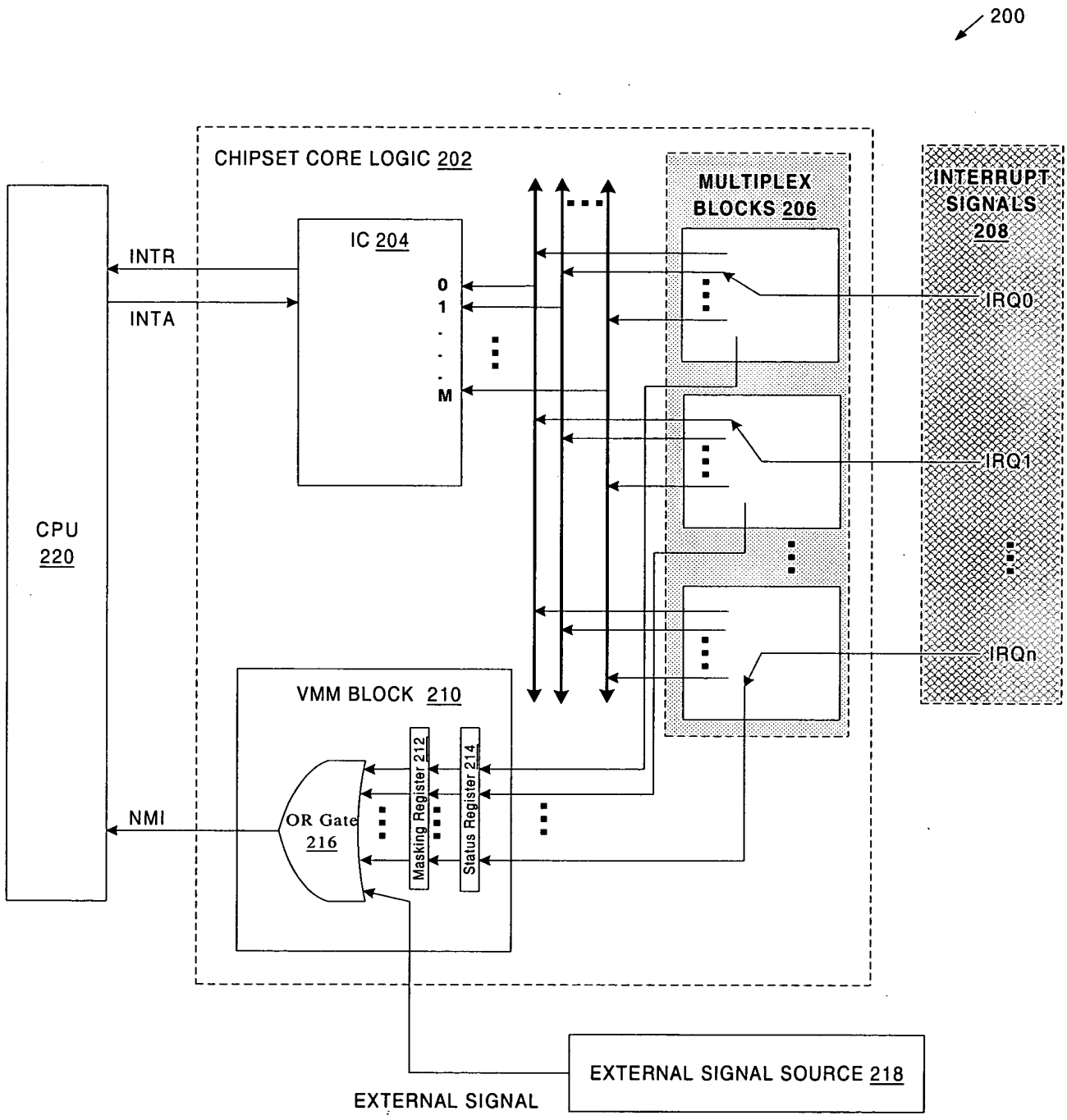
200
CHIPSET CORE LOGIC 202
MULTIPLEX BLOCKS 206
INTERRUPT SIGNALS 208
IC 204
INTR
INTA
0
1
M
IRQ0
IRQ1
IRQn
CPU 220
VMM BLOCK 210
Masking Register 212
Status Register 214
OR Gate 216
NMI
EXTERNAL SIGNAL SOURCE 218
EXTERNAL SIGNAL

FIG. 2

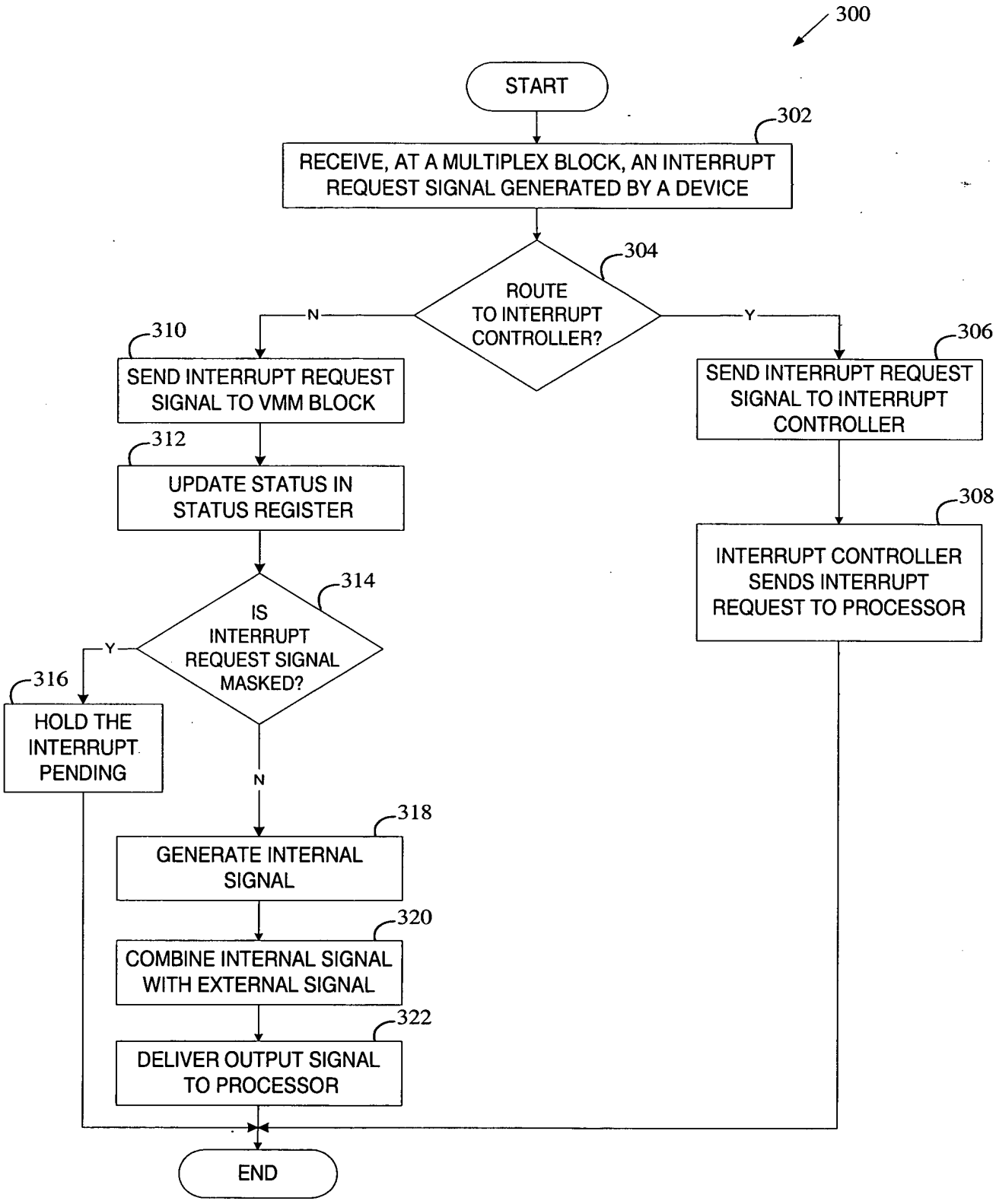
300
START
302
RECEIVE, AT A MULTIPLEX BLOCK, AN INTERRUPT REQUEST SIGNAL GENERATED BY A DEVICE
304
ROUTE TO INTERRUPT CONTROLLER?
N
Y
310
SEND INTERRUPT REQUEST SIGNAL TO VMM BLOCK
312
UPDATE STATUS IN STATUS REGISTER
314
IS INTERRUPT REQUEST SIGNAL MASKED?
Y
316
HOLD THE INTERRUPT PENDING
N
318
GENERATE INTERNAL SIGNAL
320
COMBINE INTERNAL SIGNAL WITH EXTERNAL SIGNAL
322
DELIVER OUTPUT SIGNAL TO PROCESSOR
306
SEND INTERRUPT REQUEST SIGNAL TO INTERRUPT CONTROLLER
308
INTERRUPT CONTROLLER SENDS INTERRUPT REQUEST TO PROCESSOR
END

FIG. 3

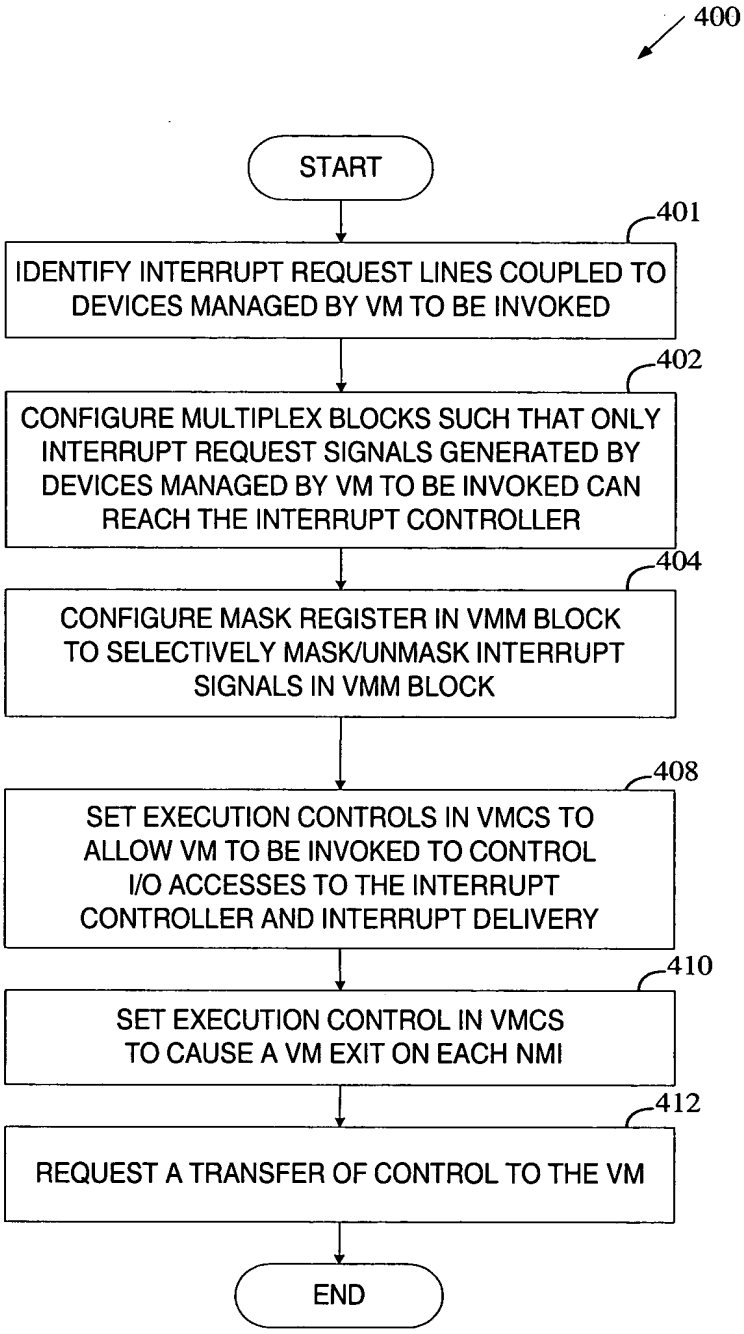
400
START
401
IDENTIFY INTERRUPT REQUEST LINES COUPLED TO DEVICES MANAGED BY VM TO BE INVOKED
402
CONFIGURE MULTIPLEX BLOCKS SUCH THAT ONLY INTERRUPT REQUEST SIGNALS GENERATED BY DEVICES MANAGED BY VM TO BE INVOKED CAN REACH THE INTERRUPT CONTROLLER
404
CONFIGURE MASK REGISTER IN VMM BLOCK TO SELECTIVELY MASK/UNMASK INTERRUPT SIGNALS IN VMM BLOCK
408
SET EXECUTION CONTROLS IN VMCS TO ALLOW VM TO BE INVOKED TO CONTROL I/O ACCESSES TO THE INTERRUPT CONTROLLER AND INTERRUPT DELIVERY
410
SET EXECUTION CONTROL IN VMCS TO CAUSE A VM EXIT ON EACH NMI
412
REQUEST A TRANSFER OF CONTROL TO THE VM
END

FIG. 4

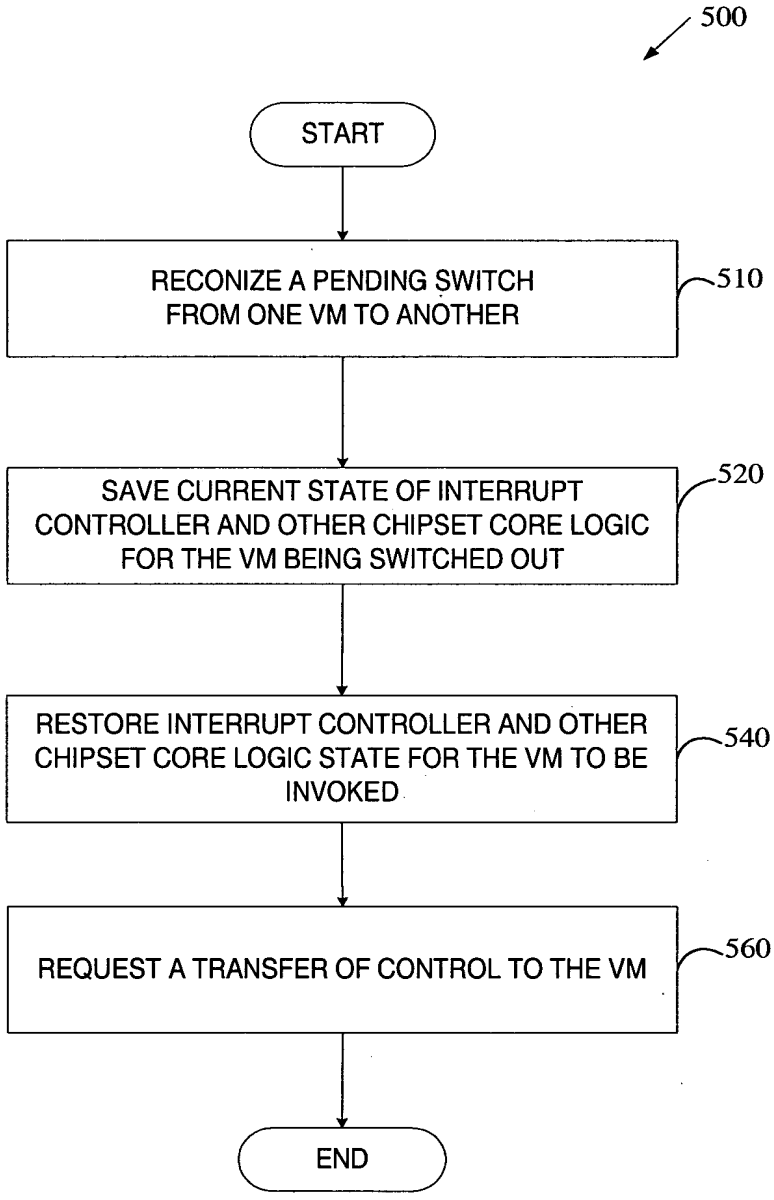
500
START
RECONIZE A PENDING SWITCH FROM ONE VM TO ANOTHER
510
SAVE CURRENT STATE OF INTERRUPT CONTROLLER AND OTHER CHIPSET CORE LOGIC FOR THE VM BEING SWITCHED OUT
520
RESTORE INTERRUPT CONTROLLER AND OTHER CHIPSET CORE LOGIC STATE FOR THE VM TO BE INVOKED
540
REQUEST A TRANSFER OF CONTROL TO THE VM
560
END

FIG. 5